REMARKS

This is in full and timely response to the Office Action mailed on October 27, 2009.

Claims 1-10 are currently pending in this application, with claims 1, 3, 5, 7, and 9 being independent.

No new matter has been added.

Reexamination in light of the following remarks is respectfully requested

New non-final Office Action

At least for the following reasons, if the allowance of the claims is not forthcoming at the very least and a new ground of rejection made, then a ***new non-final Office Action*** is respectfully requested.

Claim rejection - 35 U.S.C. §101

I. Page 2 of the Office Action contends that claims 3-4 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

This rejection is traversed at least for the following reasons.

The Commissioner now states "that computer ***programs embodied in a tangible medium***, such as floppy diskettes, are ***patentable subject matter*** under 35 U.S.C. Section 101 and must be examined under 35 U.S.C. Sections 102 and 103." *In re Beauregard*, 35 USPQ2d 1383 (Fed. Cir. 1995).

Accordingly, while not conceding the propriety of this rejection and in order to advance the prosecution of the present application, claims 3-4 have been amended.

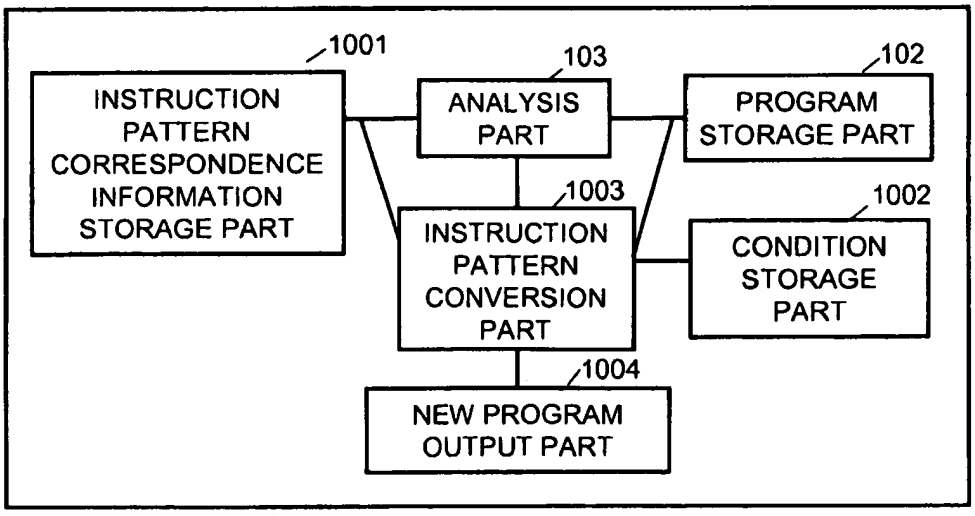
Withdrawal of this rejection and allowance of the claims is respectfully requested.

Claim rejection - 35 U.S.C. §103

II. Claims 1-10

Figure 10 of the specification as originally filed is provided hereinbelow.

FIG. 10



Figures 5, 12, and 13 of the specification as originally filed are provided hereinbelow.

FIG. 5

```
000010**********************************************************
000020* CLIENT NAME    : DENZAI PACKAGE          *
000030* PROGRAM NAME   : CLIENT BUSINESS RANKING *
000040* PROGRAM ID     : PSD712                  *
000050* CREATOR        : (I.C)                   *
000060* CREATION DATE  : 99.11.18                *
000070* AMENDMENT DATE : 00.00.00                *
000080**********************************************************
000090 IDENTIFICATION      DIVISION.
000100 PROGRAM-ID.         PSD712.
000110*
000120 ENVIRONMENT         DIVISION.
000130 CONFIGURATION       SECTION.
000140 SOURCE-COMPUTER.    V7070.
000150 OBJECT-COMPUTER.    V7070.
000160 INPUT-OUTPUT        SECTION.
000170 FILE-CONTROL.
000180*************************************
000190* CLIENT MASTER SORTED
000200*************************************
000210   SELECT  SDK30   ASSIGN  TO    K3-DK00-SDK30
000220           ORGANIZATION    SEQUENTIAL
000230           ACCESS          SEQUENTIAL
000240           SHARE           NO
000250           FILE    STATUS  STATUS-1 STATUS-2.
000260*************************************
000270* CONTROL FILE
000280*************************************
000290   SELECT  SDC00   ASSIGN  TO    C0-DK00-SDC00
000300           ORGANIZATION    RELATIVE
000310           ACCESS          DYNAMIC
000320           RELATIVE        C0-KEY
000330           SHARE           I-O
000340           FILE    STATUS  STATUS-1 STATUS-2.
```

FIG. 13

```
#ADD# PROCESS GRAPHIC CVTPICGGRAPHIC QUOTE
#ADD# IDENTIFICATION DIVISION.
000010**********************************************************
000020* CLIENT NAME    : DENZAI PACKAGE          *
000030* PROGRAM NAME   : CLIENT BUSINESS RANKING *
000040* PROGRAM ID     : PSD712                  *
000050* CREATOR        : (I.C)                   *
000060* CREATION DATE  : 99.11.18                *
000070* AMENDMENT DATE : 00.00.00                *
000080**********************************************************
#DEL# *IDENTIFICATION      DIVISION.
000100 PROGRAM-ID.         PSD712.
000110*
000120 ENVIRONMENT         DIVISION.
000130 CONFIGURATION       SECTION.
#DEL# *SOURCE-COMPUTER.    V7070.      }(1)
#CHG# SOURCE-COMPUTER.    IBM-AS400.  }(1)
#DEL# *OBJECT-COMPUTER.    V7070.      }(2)
#CHG# OBJECT-COMPUTER.    IBM-AS400.  }(2)
#ADD# SPECIAL-NAMES.
#ADD#           COPY CONVSPCN.
000160 INPUT-OUTPUT        SECTION.
000170 FILE-CONTROL.
000180*************************************
000190* CLIENT MASTER SORTED
000200*************************************
#DEL# *  SELECT  SDK30   ASSIGN  TO    K3-DK00-SDK30  ]
#CHG#    SELECT  SDK30   ASSIGN  TO    DATABASE-FCK3  ]
000220           ORGANIZATION    SEQUENTIAL
000230           ACCESS          SEQUENTIAL
#DEL# *          SHARE           NO
#DEL# *          FILE    STATUS  STATUS-1 STATUS-2.
#ADD#            FILE    STATUS  STATUS-2.
000260*************************************
000270* CONTROL FILE
000280*************************************
#DEL# *  SELECT  SDC00   ASSIGN  TO    C0-DK00-SDC00
#CHG#    SELECT  SDC00   ASSIGN  TO    DATABASE-FCC0
000300           ORGANIZATION    RELATIVE
000310           ACCESS          DYNAMIC
000320           RELATIVE        C0-KEY
#DEL# *          SHARE           I-O
#DEL# *          FILE    STATUS  STATUS-1 STATUS-2.
#ADD#            FILE    STATUS  STATUS-2.
```

FIG. 12

ID	First Instruction Pattern Information	Second Instruction Pattern Information
1	SOURCE-COMPUTER. XXXXXX	#DEL# * SOURCE-COMPUTER. XXXXXX. #CHG# SOURCE-COMPUTER. IBM-AS400.
2	OBJECT-COMPUTER. XXXXXX	#DEL# * OBJECT-COMPUTER. XXXXXX. #CHG# OBJECT-COMPUTER. IBM-AS400.
3	SELECT filexxx ASSIGN xx-DK00- ~	#DEL# *SELECT filexxx ASSIGN xx-DK00- ~ #CHG# SELECT filexxx ASSIGN DATABASE-FCxx
4	SELECT filexxx ASSIGN PR-SP00	#DEL# *SELECT filexxx ASSIGN PR-SP00 #CHG# SELECT filexxx ASSIGN PRINTER-PR
5	SELECT filexxx ASSIGN SORTWORK.	#DEL# *SELECT filexxx ASSIGN SORTWORK. #CHG# SELECT filexxx ASSIGN DATABASE-SORTWORK.
6	SELECT SCREEN ASSIGN GA-DK00 ~.	#DEL# *FORMAT-CONTROL. #DEL# * SELECT SCREEN ASSIGN GA-DK00 #DEL# * ~.
7	FORMAT SECTION.	#DEL# *FORMAT SECTION.
8	MD display name	#DEL# *MD display name.
9	WORKING-STORAGE SECTION	WORKING-STORAGE SECTION. #ADD# COPY CONVWORK.
⋮	⋮	⋮

The following description is provided for illustrative purposes and is not intended to limit the scope of the invention.

Reference is made to the specification as originally filed.

Paragraph [0060] of U.S. Patent Application Publication No. 2006/0156292, the publication document for the above-identified application, provides that:

> [0060] The instruction pattern conversion part 1003 performs a conversion process on a program stored in the program storage part 102 (an old program) so that part of the old program corresponding to the first instruction pattern information element that meets a condition stored in the condition storage part 1002 corresponds to the second instruction pattern information element that is paired with the first instruction pattern information element. The conversion part 1003 can be realized in general using an MPU and a memory. Processes required for the conversion part 1003 to convert relevant information is generally realized by software, which is stored in a memory device such as ROM. This may be, however, substituted for by hardware (using a dedicated circuit).

Paragraph [0081] of U.S. Patent Application Publication No. 2006/0156292, the publication document for the above-identified application, provides that:

> [0081] Hereinafter, the operations of the automatic program conversion device in the second embodiment will be discussed in detail. FIG. 12 shows an instruction pattern information management table stored in the instruction pattern correspondence information storage part 1001. In this table, more than one record is stacked under the common headings of "ID," "First Instruction Pattern Information," and "Second Instruction Pattern Information." IDs are used for identifying each record. This management table is used as a correspondence table when conversion takes place in search of the correspondence between the source lines in question appearing in a program as the first instruction pattern information elements, and the second

instruction pattern information elements that are individually paired with them. Specifically, referring to FIG. 12, "SOURCE-COMPUTER. XXXXXX." in the "First Instruction Pattern Information" column is converted into "#CHG# SOURCE-COMPUTER. IBM-AS400." as indicated in the "Second Instruction Pattern Information" column. Here "XXXXXX" represents any variable name, and "#DEL#" represents the deletion of the corresponding portion. "#CHG#" means that the corresponding portion is the outcome of the conversion. Moreover, lines with an asterisk (*) are comment lines, where each comment is placed between #'s.

III. U.S. Patent No. 6,901,588 (Krapf)

A. Krapf *fails* to disclose, teach, or suggest a method and apparatus *wherein at least one pair of a first instruction pattern information element representing an instruction pattern in said old source program and a second instruction pattern information element representing an instruction pattern in said new source program is stored.*

Page 3 of the Office Action contends the following:

> Krapf teaches a program conversion method, where user analyzes a source program then manually converts the source program executable in one environment to a new source program executable in other environment, (column 1, line 45-55), an expression in the old source program and its corresponding expression in the new source program are one pair of instruction pattern information

In response, FIG. 3 is a Java code fragment illustrating an example embodiment of Java components, including a Java interface Foo 202, a concrete Java class FooImpl 204, and a Java class FooUser 206 (Krapf at column 11, lines 56-59).

FIG. 4 is a C++ code fragment illustrating an example embodiment of C++ proxy components 210 wrapping Java components 200 of FIG. 3, but not having semantic usabilities closely corresponding to the semantic usabilities of the Java component of FIG. 3 (Krapf at column 12, lines 39-43).

Figures 3 and 4 of Krapf are provided hereinbelow.

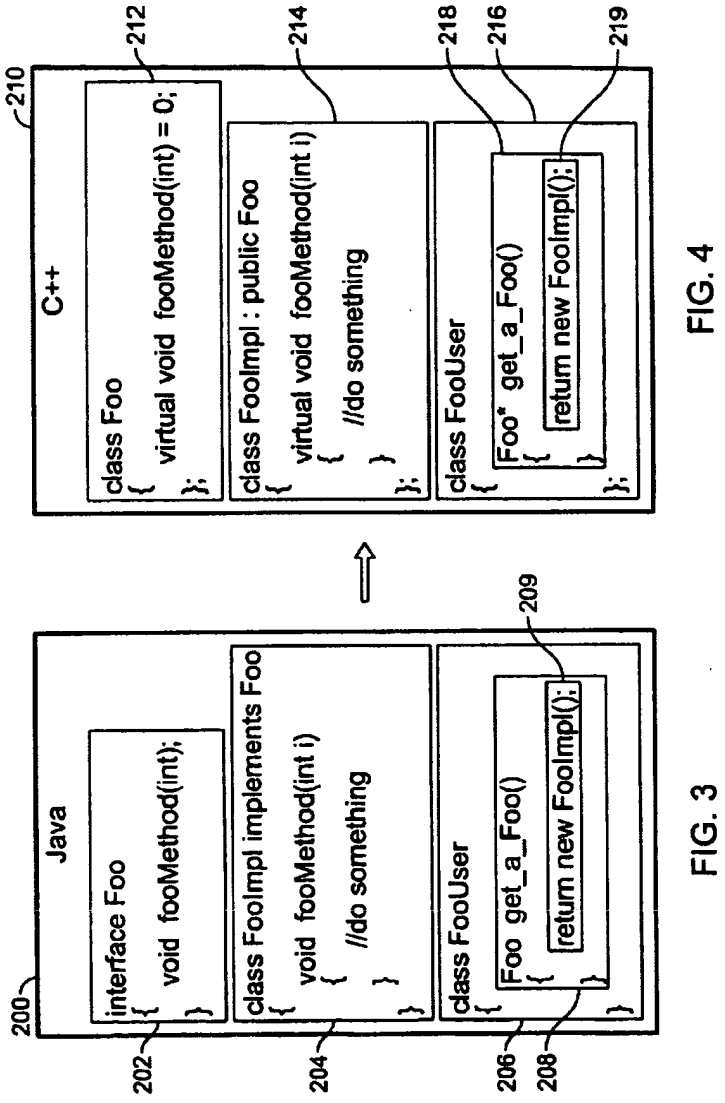


FIG. 4

FIG. 3

However, Krapf ***fails*** to disclose, teach, or suggest the ***storage of an instruction pair***, with *that pair being* the Java components 200 and the C++ proxy components 210.

As a consequence, Krapf ***fails*** to disclose, teach, or suggest a method and apparatus *wherein at least one pair of a first instruction pattern information element representing an instruction pattern in said old source program and a second instruction pattern information element representing an instruction pattern in said new source program is stored.*

B. Krapf *fails* to disclose, teach, or suggest a method and apparatus that includes *converting descriptions in an old source program that correspond to a first instruction pattern information element that appears a predetermined number of times or more, so as to correspond to a second instruction pattern information element that is paired with the first instruction pattern information element that appears predetermined number of times or more.*

Page 3 of the Office Action ***readily admits*** that Krapf ***does not teach***:

> an analysis step of analyzing, using an analysis means, a number of times that said first instruction pattern information element appears in said old source program;
>
> *an instruction pattern conversion step of converting, using a conversion means, descriptions in said old source program that correspond to said first instruction pattern information element analyzed, in said analysis step, as appearing a predetermined number of times or more, so as to correspond to said second instruction pattern information element that is paired with said first instruction pattern information element appearing said predetermined number of times or more.*

IV. U.S. Patent No. 6,044,220 (Breternitz)

Figure 12 of Breternitz is provided hereinbelow.

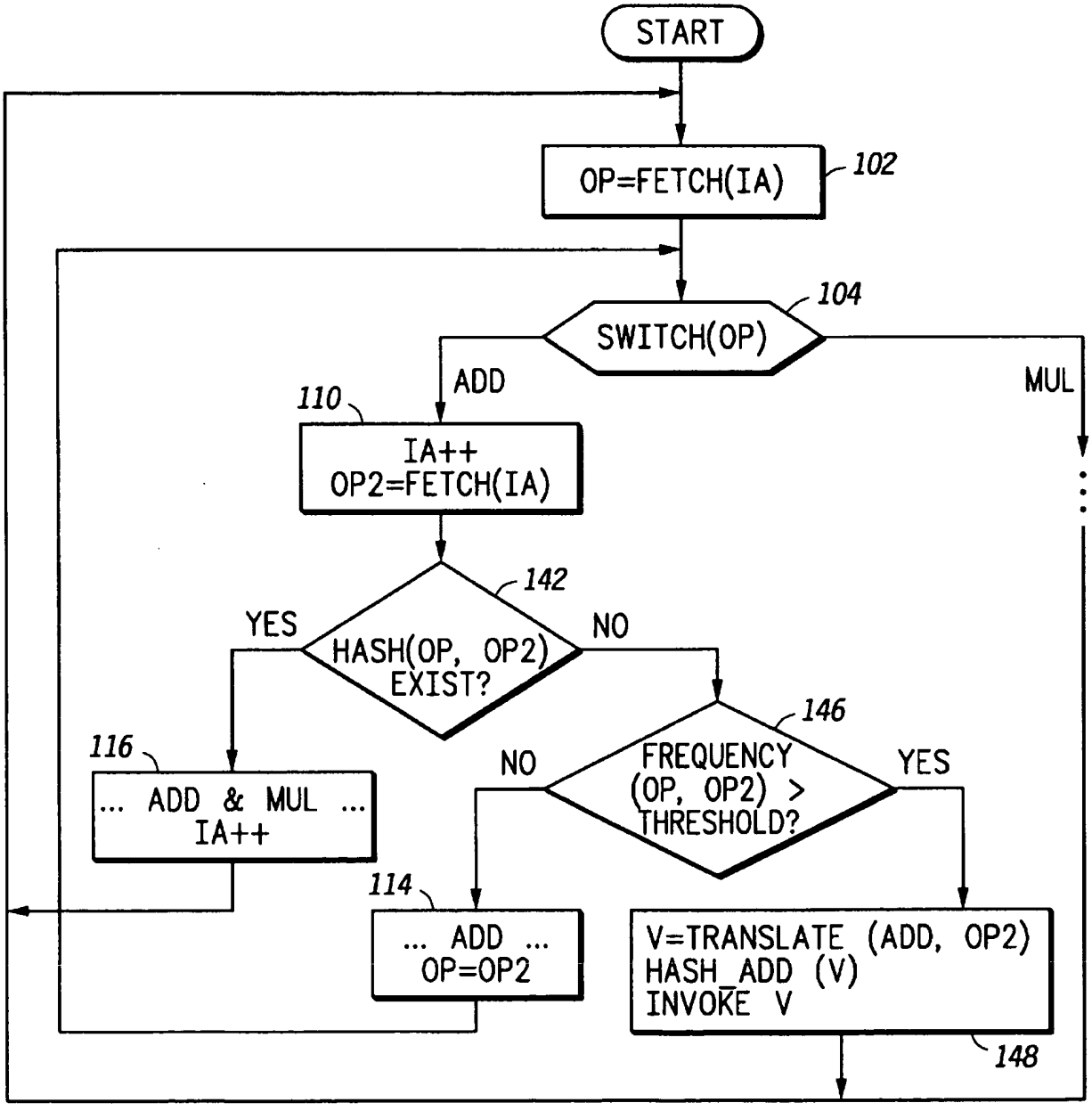


FIG.12

Breternitz in the paragraph beginning at column 5, line 62, provides that:

> FIG. 12 is a flow diagram of a dynamic hash table implementation of a turbo interpreter that uses idiom recognition. Unlike the turbo interpreter of FIG. 3, the list of recognized idioms grows dynamically. The turbo interpreter operates as a loop similar to the hash table implementation of a turbo interpreter described in FIG. 1. The primary difference shown is that when an idiom is not recognized in the hash table, step 142, a test is made of the ***frequency of occurrence of the idiom***, step 146. *If the frequency exceeds a threshold*, step 136, ***the idiom is translated into its optimized native code***, the translated code is added to the hash table, and the new idiom is executed, step 142. Otherwise, in step 114 the single "add" op code semantics are executed and the current op code is set to the second op code previously fetched in step 110. This new methodology as described is capable of recognizing idioms of length two instructions. Those skilled in the art recognize that it is a straightforward extension to recognize longer idiom sequences.

A. Breternitz *fails* to disclose, teach, or suggest a method and apparatus that includes *analyzing, using an analysis means, a number of times that said first instruction pattern information element appears in said old source program.*

Breternitz ***does not disclose*** the features that when the first instruction pattern appears more than predetermined times, the first instruction patter is converted into the second instruction pattern which is prepared and recorded in advance.

As a consequence, Breternitz fails to disclose, teach, or suggest a method and apparatus that includes analyzing, using an analysis means, a number of times that said first instruction pattern information element appears in said old source program.

B. Breternitz *fails* to disclose, teach, or suggest a method and apparatus *wherein at least one pair of a first instruction pattern information element representing an instruction pattern in said old source program and a second instruction pattern information element representing an instruction pattern in said new source program is stored.*

Breternitz arguably discloses the following in the paragraph beginning at column 3, line 6:

> According to the present invention, idioms are recognized "on-the-fly", i.e., dynamically. Dynamic idiom recognition is achieved by recognizing sequences of instruction opcodes that occur frequently and by creating special translations from them. The key idea is to extend the state machine that recognizes an opcode: when the opcode is recognized and the decoder routine is about to finish execution, it checks the following opcode and stores that information in a table. If the frequency of occurrence of that opcode exceeds a predetermined threshold, an extended idiom is created and added to the state machine. This recognition may be optimized for the failure case by assuming that, after a while, most idioms will be recognized and the state machine will account for most important idioms.

However, Breternitz ***fails*** to disclose, teach, or suggest a second instruction pattern information element paired with a first instruction pattern information element.

Specifically, the idiom disclosed in Breternitz (two opcodes) is one that is not recorded in a hush table, and is translated into a native code which is optimized to fit to the environment of converted area, and is registered in a hush table or is executed, if the idiom appears more times than a predetermined number of times. The idioms whose number of appearances is determined are not those which form pairs with the native codes and stored in a memory or the like.

As a consequence, Breternitz ***fails*** to disclose, teach, or suggest a method and apparatus *wherein at least one pair of a first instruction pattern information element representing an instruction pattern in said old source program and a second instruction pattern information element representing an instruction pattern in said new source program is stored.*

C. Breternitz *fails* to disclose, teach, or suggest a method and apparatus that includes *converting descriptions in an old source program that correspond to a first instruction pattern information element that appears a predetermined number of times or more, so as to correspond to a second instruction pattern information element that is paired with the first instruction pattern information element that appears predetermined number of times or more.*

Page 4 of the Office Action asserts that: *Breternitz teaches analyzing a source program, and determine the occurrence of each instruction in the source program, if the occurrence of an instruction exceeds certain threshold, then the instruction is translated automatically to a corresponding instruction executable in another environment.*

In response, Breternitz arguably discloses that *dynamic **idiom** recognition* is achieved by ***recognizing sequences of instruction opcodes*** that occur frequently and by creating special translations from them (Breternitz at column 3, lines 7-10).

However, Breternitz ***fails*** to disclose, teach, or suggest a second instruction pattern information element paired with a first instruction pattern information element.

Specifically, Breternitz ***fails*** to disclose, teach, or suggest the optimized native code *being **paired with*** the idiom.

Breternitz arguably discloses that the primary difference shown is that when an idiom is not recognized in the hash table, step 142, a test is made of the ***frequency of occurrence of the***

idiom, step 146. *If the frequency exceeds a threshold*, step 136, ***the idiom is translated into its optimized native code***, the translated code is added to the hash table, and the new idiom is executed, step 142 (Breternitz at column 5, line 67 to column 6, line 3).

However, the translation is in the manner that opcode (operation code) is translated into native code, which is different from the originally prepared or stored code prior to the translation.

Thus, Breternitz does not convert a source code in an environment into the other source code in a different environment and cannot be used to the conversion of a source program.

Additionally, the translation of Breternitz is not the conversion to a native code which is prepared in advance in connection with idiom.

As a consequence, Breternitz ***fails*** to disclose, teach, or suggest a method and apparatus that includes *converting descriptions in an old source program that correspond to a first instruction pattern information element that appears a predetermined number of times or more, so as to correspond to a second instruction pattern information element that is* ***paired with*** *the first instruction pattern information element that appears predetermined number of times or more.*

D. Breternitz *fails* to disclose, teach, or suggest a method and apparatus that includes *an input of manually entered by a user, regarding descriptions in said old source program that correspond to said first instruction pattern information element analyzed, in said analysis part, as appearing less than said predetermined number of times, so that said descriptions will be modified for said new source program.*

Breternitz does not disclose, teach or suggest counting numbers of appearances of the first instruction pattern information that form a pair with the second instruction pattern information stored in a memory, and, if the number exceeds the predetermined number, the old source program

description part that corresponds to the first instruction pattern information is converted so as to correspond to the second instruction pattern information.

As a consequence, Breternitz ***fails*** to disclose, teach, or suggest a method and apparatus that includes *an input of manually entered by a user, regarding descriptions in said old source program that correspond to said first instruction pattern information element analyzed, in said analysis part, as appearing less than said predetermined number of times, so that said descriptions will be modified for said new source program.*

V. U.S. Patent No. 7,086,046 (Barsness)

A. Barsness ***fails*** to disclose, teach, or suggest a method and apparatus that includes *converting descriptions in an old source program that correspond to a first instruction pattern information element that appears a predetermined number of times or more, so as to correspond to a second instruction pattern information element that is paired with the first instruction pattern information element that appears predetermined number of times or more.*

Figures 3A-3H of Barsness depict a user interface 302 for displaying various examples of compiler-optimized code on the output device 128, e.g., a display device (Barsness at column 5, lines 28-30).

However, Barsness ***fails*** to disclose, teach, or suggest a second instruction pattern information element paired with a first instruction pattern information element.

Withdrawal of this rejection and allowance of the claims is respectfully requested.

Official Notice

There is no concession as to the veracity of Official Notice, if taken in any Office Action.

An affidavit or document should be provided in support of any Official Notice taken. 37 C.F.R. §1.104(d)(2), M.P.E.P. §2144.03. See also, *Ex parte Natale*, 11 USPQ2d 1222, 1227-1228 (Bd. Pat. App. & Int. 1989)(failure to provide any objective evidence to support the challenged use of Official Notice constitutes clear and reversible error).

Extensions of time

Please treat any concurrent or future reply, requiring a petition for an extension of time under 37 C.F.R. §1.136, as incorporating a petition for extension of time for the appropriate length of time.

The Commissioner is hereby authorized to charge all required fees, fees under 37 C.F.R. §1.17, or all required extension of time fees.

Fees-general authorization

The Commissioner is hereby authorized to charge any deficiency in fees filed, asserted to be filed, or which should have been filed herewith (or with any paper hereafter filed in this application by this firm).

If any fee is required or any overpayment made, the Commissioner is hereby authorized to charge the fee or credit the overpayment to Deposit Account # 18-0013.

Conclusion

This response is believed to be a complete response to the Office Action.

Applicants reserve the right to set forth further arguments supporting the patentability of their claims, including the separate patentability of the dependent claims not explicitly addressed herein, in future papers.

For the foregoing reasons, all the claims now pending in the present application are allowable, and the present application is in condition for allowance.

Accordingly, favorable reexamination and reconsideration of the application in light of the remarks is courteously solicited.

If the Examiner has any comments or suggestions that could place this application in even better form, the Examiner is requested to telephone Brian K. Dutton, Reg. No. 47,255, at 202-955-8753.

Dated: March 23, 2010

Respectfully submitted,

By_____
Brian K. Dutton
Registration No.: 47,255
RADER, FISHMAN & GRAUER PLLC
Correspondence Customer Number: 23353
Attorney for Applicant